

AMENDMENTS TO THE CLAIMS

The listing of claims below replace all prior versions, and listings, of claims:

A 1 1. (Original) A method of performing a transaction in a database system,
2 comprising:

B 3 receiving a transaction to be performed, wherein the transaction is
4 processed by a plurality of access modules; and

5 performing a flush of a transaction log in each access module before an
6 end transaction procedure.

1 2. (Currently Amended) The method of claim 1, further comprising issuing a
2 request to flush the transaction log with a message sent to each access module for
3 performing a last step of the transaction, the last step performed prior to the end
4 transaction procedure.

1 3. (Currently Amended) The method of claim 2, further comprising avoiding
2 performing the flush of the transaction log in a data access step prior to the end
3 transaction procedure to avoid performance of a transaction log flush in the end
4 transaction procedure.

1 4. (Original) The method of claim 2, further comprising determining that the
2 last step is being performed by all of the plurality of access modules.

1 5. (Original) The method of claim 1, further comprising determining if the
2 transaction log has been flushed before performing the end transaction procedure.

1 6. (Original) The method of claim 5, further comprising avoiding
2 performance of a transaction log flush in the end transaction procedure if the transaction
3 log has been flushed.

1 7. (Original) The method of claim 1, further comprising:
2 identifying the transaction as an implicit transaction.

AT
B
1 8. (Original) The method of claim 1, further comprising:
2 performing the end transaction procedure, which follows execution of the
3 transaction.

1 9. (Original) The method of claim 8, performing the end transaction
2 procedure comprising:

3 skipping broadcast of a directive indicating commencement of the end
4 transaction procedure to the plurality of access modules.

1 10. (Original) A method of performing an end transaction procedure in a
2 database system, comprising:

3 a first access module in the database system writing an end transaction
4 indication to a first transaction log portion, the first access module being part of a
5 cluster of access modules; and

6 the first access module sending an end transaction directive to a
7 fallback module associated with the first access module, the fallback module being
8 part of the cluster.

1 11. (Original) The method of claim 10, wherein the first access module sends
2 the end transaction directive to the fallback module but not to other access modules in the
3 cluster.

1 12. (Original) The method of claim 10, wherein sending the end transaction
2 directive comprises sending an end transaction-part one directive.

1 13. (Original) The method of claim 12, further comprising the first access
2 module broadcasting an end transaction-part two directive to all access modules in the
3 cluster.

AT 1 14. (Original) The method of claim 10, further comprising the fallback
2 module writing an end transaction indication to a second transaction log portion.

B { 1 15. (Original) The method of claim 10, further comprising the first access
2 module flushing the first transaction log portion.

1 16. (Original) The method of claim 10, further comprising the first access
2 module flushing the first transaction log portions but the other access modules in the
3 cluster not flushing their respective transaction log portions.

1 17. (Original) A database system comprising:
2 a plurality of storage media; and
3 a plurality of access modules, wherein each access module is coupled to
4 one of the plurality of storage media; and
5 each of the access modules being adapted to flush a transaction log before
6 performing an end transaction procedure.

1 18. (Original) The database system of claim 17, further comprising a
2 controller adapted to determine if each access module has flushed the transaction log
3 maintained by the access module.

1 19. (Original) The database system of claim 18, wherein the controller is
2 adapted to skip sending a directive to perform a transaction log flush if the controller
3 determines that each access module has flushed the transaction log before the end
4 transaction procedure.

1 20. (Original) The database system of claim 17, further comprising a
2 controller adapted to provide a flush directive with a message to each of the access
3 modules to perform a last step of the transaction.

AT
B1.

1 21. (Original) An article comprising a medium storing instructions for
2 enabling a processor-based system to:
3 receive a transaction to be performed, wherein the transaction is processed
4 by a plurality of access modules ;
5 determine that a last step of the transaction involves the plurality of access
6 modules; and
7 flush a transaction log to a storage while the last step is performed by the
8 plurality of access modules.

1 22. (Original) The article of claim 21, further storing instructions for enabling
2 the processor-based system to:
3 perform an end transaction, wherein the end transaction follows execution
4 of the transaction.

1 23. (Original) The article of claim 22, further storing instructions for enabling
2 a processor-based system to:
3 avoid broadcast of a directive indicating commencement of the end
4 transaction to the plurality of access modules.

1 24. (Original) A method of performing a transaction in a database system,
2 comprising:
3 receiving a transaction to be performed on plural access modules in the
4 database system;
5 maintaining a log to track operations performed in the transaction;
6 writing the log to persistent storage before start of an end transaction
7 procedure.

1 25. (Original) The method of claim 24, wherein writing the log to persistent
2 storage comprises flushing the log.

1 26. (Original) The method of claim 24, wherein maintaining the log comprises
2 maintaining a transaction log.

1 27. (Original) The method of claim 24, further comprising performing the end
2 transaction procedure, the end transaction procedure comprising writing an end
3 transaction indication into the log.

1 28. (Original) A database system comprising:
2 storage media;
3 access modules coupled to the storage media; and
4 a parsing engine coupled to the access modules, the parsing engine
5 adapted to perform one of:

6 (a) providing a directive with a message to perform a last step
7 of a transaction and communicating the directive to the access modules, each access
8 module responsive to the directive to perform a transaction log flush before performance
9 of an end transaction procedure; and

10 (b) determining if each of the access modules has performed a
11 transaction log flush before start of the end transaction procedure;
12 the parsing engine adapted to avoid sending a broadcast directive to the
13 access modules to cause performance of a transaction log flush during the end transaction
14 procedure.

1 29. (New) The method of claim 1, wherein the transaction comprises plural
2 steps, the method further comprising:

3 performing the plural steps prior to performing the end transaction
4 procedure, and

5 wherein performing the flush of the transaction log comprises performing
6 the flush of the transaction log in one of the plural steps.

1 30. (New) The method of claim 29, wherein performing the plural steps
2 comprises performing, in each of the plural steps, access of relational table data stored in
the database system.

A23
B1
1 31. (New) The method of claim 30, wherein performing the flush of the
2 transaction log in one of the plural steps comprises performing the flush of the transaction
3 log in a last one of the plural steps.

1 32. (New) The method of claim 31, further comprising each access module
2 adding a first entry to the transaction log to redo the transaction by the access module in
3 case of system failure.

1 33. (New) The method of claim 4, wherein performing the flush of the
2 transaction is prior to the end transaction procedure if the last step is performed by all of
3 the plurality of access modules, the method further comprising:

4 performing the flush of the transaction log in the end transaction
5 procedure if the last step is not performed by all of the plurality of access modules.

1 34. (New) The database system of claim 17, wherein the access modules to
2 perform a transaction comprising plural steps, one or more of the access modules adapted
3 to perform the plural steps prior to the end transaction procedure, and the access modules
4 adapted to perform the flush of the transaction log in one of the plural steps.

1 35. (New) The database system of claim 34, wherein the one of the plural
2 steps comprises a last one of the steps.

1 36. (New) The database system of claim 35, wherein the transaction log
2 comprises a first entry associated with each access module to enable a redo of the
3 transaction in case of system failure.

1 37. (New) The database system of claim 36, wherein the transaction log
2 further comprises a second entry associated with each access module to enable an undo of
the transaction.

AJ 3
B1 1 38. (New) The database system of claim 34, further comprising a controller to
2 determine whether a last one of the steps involves all the access modules, and in response
3 to determining that the last one of the steps involves all the access modules, the controller
4 to send a directive to all the access modules to perform the flush of the transaction log in
5 the last one of the steps.

1 39. (New) The database system of claim 38, in response to determining that
2 the last step does not involve all access modules, the controller to send a directive to
3 perform the flush of the transaction log in the end transaction procedure.

1 40. (New) The article of claim 21, wherein the transaction comprises plural
2 steps, the article further storing instructions for enabling a processor-based system to:
3 perform the plural steps prior to performing the end transaction procedure,
4 and

5 wherein performing the flush of the transaction log comprises performing
6 the flush of the transaction log in one of the plural steps.

1 41. (New) The article of claim 40, wherein performing the plural steps
2 comprises performing, in each of the plural steps, access of relational table data stored in
3 the database system.

1 42. (New) The article of claim 41, wherein performing the flush of the
2 transaction log in one of the plural steps comprises performing the flush of the transaction
3 log in a last one of the plural steps.

AZ
1
B1
2
3

43. (New) The article of claim 42, further storing instructions for enabling a processor-based system to cause each access module to add a first entry to the transaction log to redo the transaction by the access module in case of system failure.
